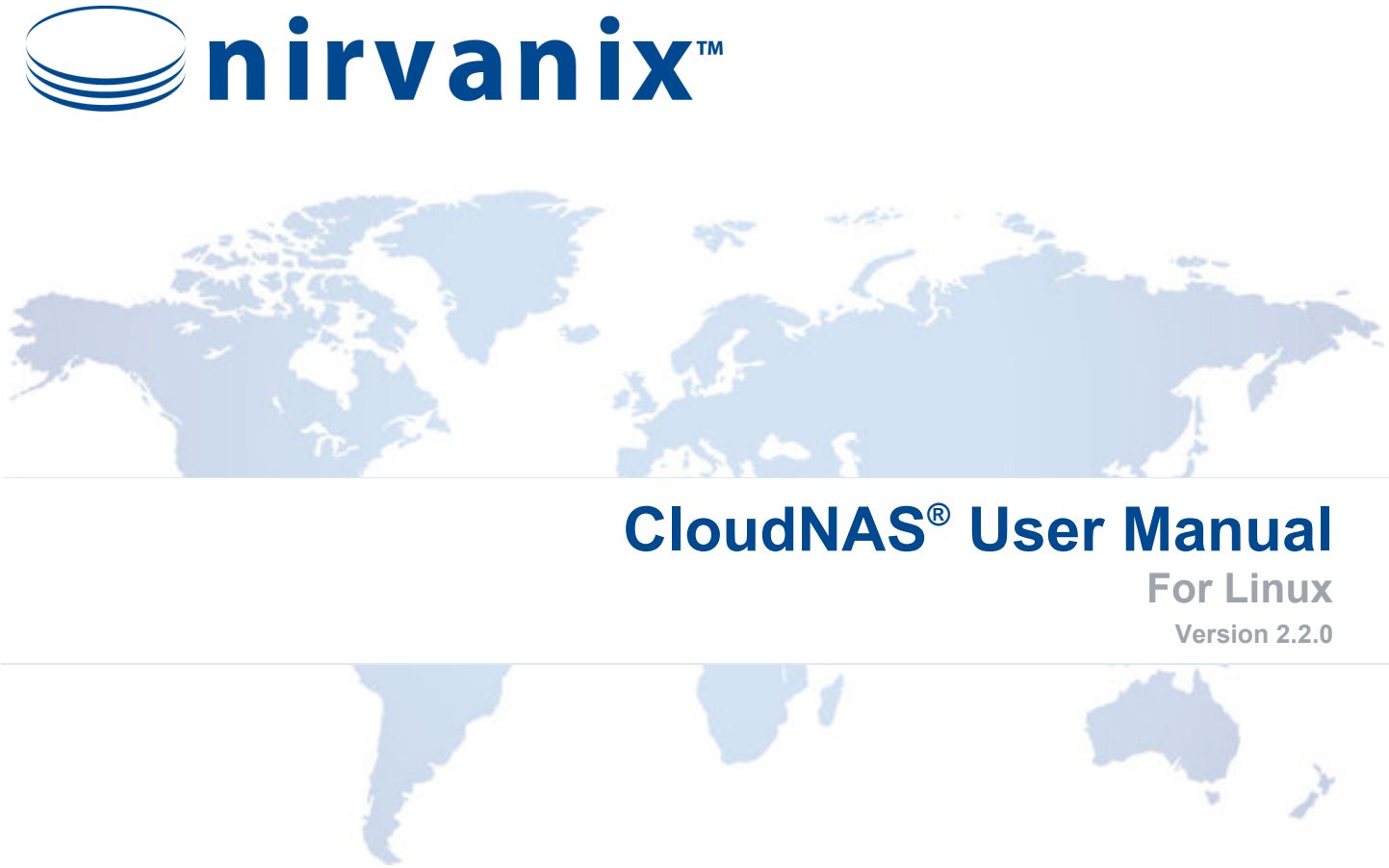

CloudNAS® User Manual

For Linux

Version 2.2.0

Neither this document nor any part of it may be copied or reproduced in any form or by any means without the prior written consent of Nirvanix, Inc. The information contained in this document is subject to change or revision without prior notice. © Copyright 2008 – 2010 Nirvanix, Inc. All rights reserved.

Table of Contents

1. Overview 3
2. System Requirements 3
3. Product Activation 4
4. Installation Instructions 5
4.1 Required Software 5
4.2 Prerequisite Installation 5
4.3 Installation 6
4.4 Mount / Unmount / Auto Run 6
4.5 Accessing Your Mounted File System 7
4.6 Uninstallation 8
5. Configuration 9
5.1 Upload Monitor 9
5.2 Buffer & Cache 9
5.3 Encryption 10
5.4 Automatic Software Updates 11
5.5 Custom Configuration 12
6. Modules and Functional Descriptions 15
6.1 Upload Buffer & Read Cache 15
6.2 Cache Disabled and Write Operations 15
6.3 Logging 15
6.4 File Ownership 16
6.5 File Permissions 16
7. Known Issues 17
8. Troubleshooting 19
8.1 Activation 19
8.2 Performance 19
8.3 File Access 19
8.4 Mounting / Unmounting 20
Appendix I – NFS Setup 21
Appendix II – Advanced Configuration 22

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

1. Overview

Nirvanix CloudNAS® for Linux provides local file system access to a Nirvanix Storage Account. This creates a seamless, secure Cloud Storage target for many backup applications.

Nirvanix CloudNAS is built upon the FUSE (Filesystem in Userspace) project. FUSE acts as an adapter between the local file system and the Nirvanix Web Services.

2. System Requirements

- Server-class hardware with minimum 2 GHz processor, 2 GB RAM

  Cache Enabled (default)
  The greater of four times the largest single set of files that will be copied to CloudNAS at the same time or a minimum of 200 GB.
  Note: Cache must be enabled for use of CloudNAS encryption.

  Cache Disabled
  100 GB free disk space recommended.

- Red Hat Enterprise Linux 5.0 (32- or 64-bit); CentOS 5 (32-bit); SUSE 10 (32- or 64-bit); Ubuntu 9 (32- or 64-bit)
  Support for additional operating systems may be added by request. Please contact Nirvanix Customer Support for details.

- FUSE 2.7.3

- Unfs3 – *Only if exported as an NFS share. See Appendix I for details*.

- Samba 3.3.4 – *Only if exported as a CIFS share. Note: Previous versions of Samba (including those that may come pre-installed on Red Hat Enterprise Linux and other distributions) will **not** work correctly with CloudNAS.*

- High-speed internet connection
  Ports 80 (HTTP) and 443 (HTTPS) must be open and accessible for the domains listed below. These settings may need to be applied to corporate firewalls and/or proxies:
  - services.nirvanix.com
  - node*X*.nirvanix.com – "*X*" is a numeric designation of the nodes in the Nirvanix SDN currently covering "1" though "5"
  - Any Virtual URLs that have been set-up for your account in the Nirvanix SDN

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

3. Product Activation

To use CloudNAS, your Nirvanix Account must be enabled to allow application activation and ensure proper update notifications. If your account is not CloudNAS enabled or if you are not sure, contact Nirvanix support to confirm the status of your account.

If the version of Nirvanix CloudNAS that you are running has an update, the technical contact for your account will receive a notification. On occasion Nirvanix may issue mandatory updates for CloudNAS.

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

4. Installation Instructions

The installation of Nirvanix CloudNAS requires a few steps that will allow you to get started quickly and easily. To complete the installation, please have the following information on hand:

a) Account username

b) Account password

c) Application Name

d) Application Key

This section discusses how to complete these prerequisites and get started with the installation. The prerequisites can be different depending on the type of distribution you are using or pre-existing software that has been installed.

4.1 Required Software

Nirvanix CloudNAS requires FUSE (Filesystem in Userspace) to be installed. During its own installation process, Nirvanix CloudNAS will verify that the required software is installed properly.

FUSE is available from its official project site:
http://fuse.sourceforge.net/

Please install the hosted package, FUSE 2.7.3:
http://downloads.sourceforge.net/fuse/fuse-2.7.3.tar.gz?big_mirror=0

4.2 Prerequisite Installation

These steps have been tested with a base install and will result in preparing your system to be used with Nirvanix CloudNAS.

1. Download the recommended FUSE kernel modules and utilities.

2. Install FUSE

    a) *Option1:*
    download dkms-fuse rpms
    rpm –i dkms-fuseXXXX.rpm

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

b) *Option 2 (Requires gcc and make):*
Extract FUSE tar.gz using tar –xzf fuseXXXX.tar.gz
cd fuseXXXXX
./configure
make
make install
modprobe fuse

4.3 Installation

To install Nirvanix CloudNAS you should execute the following steps.

1. Extract the contents of “cloudnas.tar.gz”.
2. Copy the cloudnas directory to “/opt/”.
3. Run `/opt/cloudnas/cloudnas-install` to install the binaries and validate the dependencies.
4. Run `/opt/cloudnas/cloudnas-config` to configure and activate your install with a Nirvanix Master Account.
5. Run either `mount.cloudnas` or start the daemon “/etc/init.d/cloudnasd”.

Note: If you are installing Nirvanix CloudNAS on a virtual machine be aware of the system storage requirements when creating your virtual disks.

4.4 Mount / Unmount / Auto Run

To mount Nirvanix CloudNAS you should execute the following steps.

1. Run either `mount.cloudnas` or start the daemon “/etc/init.d/cloudnasd”.

To unmount Nirvanix CloudNAS you should execute the following steps.

1. Run `uploadmon.cloudnas` to verify that all data has finished transferring to the SDN.
2. Once verified, run either `unmount.cloudnas` or stop the daemon “/etc/init.d/cloudnasd”.

To set Nirvanix CloudNAS to auto run following a reboot you should execute the following steps if using a Red Hat distribution:

1. If you have installed FUSE through the rpm (see “Prerequisite Installation” above), FUSE will load automatically upon reboot. If you choose to compile your FUSE module, you will

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

need to add “modprobe fuse” (no quotes) to the /etc/rc.d/rc.sysinit script.

2. Determine the current run level of your operating system: enter “runlevel” at the command prompt, for example:

```
[root@localhost ~]# runlevel

N 5
```

3. Now, run the following command:

```
chkconfig --level {runlevel} cloudnasd on
```

This will automatically create a symbolic link in the /etc/rc.d/rc.{runlevel} directory to the /etc/init.d/cloudnasd startup script. For example:

```
chkconfig --level 5 cloudnasd on
```

4. CloudNAS should now start automatically upon reboot.

For SUSE distributions execute the following steps:

1. It is recommended to install FUSE through the RPM methodology noted in the “Prerequisite Installation” section above. If installed in this manner, FUSE will automatically load upon reboot. If you do not choose this methodology, you can add fuse to the MODULES_LOADED_ON_BOOT section in /etc/sysconfig/kernel.
2. Follow steps 2 through 4 for Red Hat distributions documented above.

4.5 Accessing Your Mounted File System

After installation is successfully completed, your mounted files are accessible at “/nirvanix/[Nirvanix Account Name]”.

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

4.6 Uninstallation

After uninstalling Nirvanix CloudNAS, any files uploaded to the SDN will remain on the SDN. If you need to delete or copy files already on the SDN, this should be completed prior to uninstalling the CloudNAS. If you do not, you may need to reinstall the CloudNAS to do so.

To uninstall Nirvanix CloudNAS you should execute the following steps:

1. Run `uploadmon.cloudnas` to verify that all data has finished transferring to the SDN.
2. Once verified, run either `unmount.cloudnas` or stop the daemon using "/etc/init.d/cloudnasd stop".
3. Run `uninstall.cloudnas` to remove all local files associated with the CloudNAS.

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

5. Configuration

The Information here will help you configure and use your Nirvanix CloudNAS.

5.1 Upload Monitor

If Nirvanix CloudNAS is running with caching enabled, the status of file uploads may be tracked using the upload monitor. This is done as follows:

1. Confirm Nirvanix CloudNAS is mounted and accessible (described in the section above).
2. Run the command `uploadmon.cloudnas`

The upload monitor will output the number of files uploading, the local disk path and remote Nirvanix path of each file uploading, and the bytes remaining to upload.

If the command line argument **–completed** or **–c** is specified, the output will be specific to completed uploads. The remote Nirvanix path, file size, upload time in seconds, along with basic upload performance for the files transferred to the SDN since the CloudNAS installation was started. Restarting CloudNAS will reset the upload performance metric.

5.2 Buffer & Cache

Nirvanix CloudNAS is configured to run with caching enabled when initially installed. The default directory containing the cache is “/usr/share/nirvanix”. The default cache size will use 25% of the available space on the designated partition. These settings can be changed to a different percentage or to a specific amount designated in MBs (megabytes). Information on changing these settings can be found in the Cache & Buffer section of Custom Configuration.

The cache also holds directory information to enable faster data access and improve performance. If you are using more than one CloudNAS to access the same storage repository it is possible for the cache to become out of sync. This increases the risk of collisions if one CloudNAS is overwriting a file as another is still updating it. The same problem may be seen if accessing the same repository through any other interface other than the CloudNAS. There are configuration changes that can be made to minimize this risk however they do create some tradeoffs on performance. Please contact the Nirvanix support team for more information.

Configuration options related to buffer & cache are further detailed in Section 5.5.

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

5.3 Encryption

As an added security feature, Nirvanix CloudNAS supports the encryption of data for storage on the SDN. This feature uses industry standard AES and supports up to 256-bit keys that are user generated. This is separate from and can be used in addition to an optional encrypted Secure Sockets Layer (SSL) connection which secures the transfer of data from the CloudNAS to the SDN.

To enable encryption you will need to create the following settings in the “cloudnas.conf” file. Please see the “Custom Configuration” section of this manual for additional details:

- Encryption.Enable=True
- Encryption.Key.Path=etc/encryption.key [or you can designate an alternate location]
- Encryption.InitialVector.Path=etc/encryption.iv [or you can designate an alternate location]

IMPORTANT: Nirvanix cannot help decrypt data. Please store encryption keys, chunk sizes, and initial vector information in a safe place and manage it appropriately.

IMPORTANT: When using CloudNAS encryption, the software encrypts and stores files in fixed-size chunks for efficient access. As a result, data encrypted using CloudNAS can only be decrypted with an installation using the same chunk size, encryption key, and initial vector. Other methods of accessing data on the SDN will not be able to decrypt data.

Some additional information to know about the Nirvanix CloudNAS encryption feature:

- Changing the encryption keys in CloudNAS does not update the encryption of data already transferred to the SDN. You must download and decrypt the data using the old keys and then re-upload and encrypt it again you if would like the keys to be the same.
- Encryption is only supported if the CloudNAS buffer cache is enabled. Enabling encryption automatically overrides file buffer cache settings enabling buffer caching as well.
- Data corruption will occur if you attempt to modify or edit CloudNAS encrypted data using CloudNAS with cache disabled.
- Enabling encryption may negatively affect upload performance.
- The Key file must be 16 bytes (128 bits), 24 bytes (192 bits), or 32 bytes (256 bits).

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

- The Initial Vector file must be 16 bytes (128 bits).
- Use of encryption is not compatible with SDN features such as search, transcode, and image manipulation.
- The encryption process is applied to the data stream on the local machine as it is uploaded to the SDN. Data in the Nirvanix CloudNAS cache on the local machine is not encrypted.

5.4 Automatic Software Updates

CloudNAS periodically checks with the Nirvanix SDN to see if there is a new version available and has the ability to update itself to the latest recommended release. This is a configurable setting and has the following options:

1. Notify – Notify the account technical contact of an available update.
2. NotifyDownload **[Default setting]** – Notify the account technical contact of an available update and download the update so that an administrator can run it manually.
3. NotifyDownloadUpdate – Notify the account technical contact of an available update, download the update, and automatically run the installation/upgrade process based on the day/time setting configured for the CloudNAS instance.

Only one notification is sent per available update regardless of the number of CloudNAS instances running under that account.

Nirvanix classifies new releases into two groups: 1) Recommended/Critical; 2) Mandatory. Virtually all updates will fall into the first group and will cover simple patches as well as incremental version and functionality releases. These are handled as part of the normal update process. The second group is reserved for situations where a specific version of CloudNAS is found to cause data inconsistencies or undue impacts to the SDN. In this situation, deployed instances of that version will no longer be able to synchronize with the SDN and the installation must be updated to restore normal CloudNAS operations.

Automatic updates must be configured to run on a specific day of the week at a set time. This is to allow administrators to set the installation to coincide with recurring maintenance windows that are already in use.

Configuration options related to automatic software updates can be found in Section 5.5.

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

5.5 Custom Configuration

The configuration file has a number of different values that will let you change the overall behavior of Nirvanix CloudNAS and allow specific settings to be changed. Each time you change the configuration you will need to restart the service or remount the drive.

Misspelling settings or values and incorrect case usage will cause the setting to be quietly ignored. The same is true if extra characters are included in the configuration setting line.

Configuration settings are placed into /opt/cloudnas/etc/cloudnas.conf. Each setting is a separate line with the format: setting=value.

Example of settings as they appear in “cloudnas.conf”:

```
Log.Syslog.Enable=true
Log.Level=Warning
```

Below is a formatted list of configuration values and their meanings:

Encryption

Setting: Encryption.Enable
Default Value: false
Definition: Enables encryption when true. If cache is disabled the encryption process fails upon configuration and the CloudNAS will exit with an error message.
Example: Encryption.Enable=false

Setting: Encryption.Key.Path
Default Value: etc/encryption.key
Definition: The path containing the key for encryption purposes.
Example: Encryption.Key.Path=etc/encryption.key

Setting: Encryption.InitialVector.Path
Default Value: etc/encryption.iv
Definition: The path containing the seed value for the encryption algorithm.
Example: Encryption.InitialVector.Path=etc/encryption.iv

Web Services

Setting: Nirvanix.WS.SecureFileTransfer
Default Value: false
Definition: Forces all calls to use an encrypted SSL connection over HTTP when true. If false, only authentication calls use an encrypted SSL connection.
Example: Nirvanix.WS.SecureFileTransfer=false

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

Logging

Setting: Log.Syslog.Enable
Default Value: true
Definition: When true the file system will log to syslog.
Example: Log.Syslog.Enable=true

Setting: Log.Level
Default Value: Info
Definition: The level of logging verbosity. Valid values are "Error", "Warning", and "Info".
Example: Log.Level=Info

Software Updates

Setting: SoftwareUpdate.DownloadDirectory
Default Value: The current working directory of the CloudNAS installation.
Definition: The local path to where new software versions are automatically downloaded.
Example: SoftwareUpdate.DownloadDirectory=/path/to/CloudNAS

Setting: SoftwareUpdate.Policy
Default Value: NotifyDownload
Definition: The policy to use when applying a software update. Valid values are "Notify", "NotifyDownload", or "NotifyDownloadUpdate".
Example: SoftwareUpdate.Policy=NotifyDownload

Setting: SoftwareUpdate.Policy.UpdateSchedule
Default Value: null
Definition: This is required and only applicable when SoftwareUpdate.Policy is set to "NotifyDownloadUpdate". The format is as follows and is not case sensitive:

```
      Daily.[time]          i.e.  Daily.14:30
      Weekly.[day].[time]   i.e.  Weekly.Tuesday.22:10
```

Example: SoftwareUpdate.Policy.UpdateSchedule=Daily.14:30

Permissions & Ownership

Setting: Defaults.Group
Default Value: 0
Definition: The default group for files without ownership assigned. This value may be a group name or id.
Example: Defaults.Group=0

Setting: Defaults.Owner
Default Value: 0
Definition: The default owner for files without ownership assigned. This may be a user name or id.
Example: Defaults.Owner=0

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

Setting: Defaults.EnableKernelPermissionChecking
Default Value: true
Definition: When true the OS will enforce all permissions, otherwise permissions will be ignored.
Example: Defaults.EnableKernelPermissionChecking=true

Setting: Defaults.Name
Default Value: nirvanix
Definition: The name to report for the file system.
Example: Defaults.Name=nirvanix

Buffer & Cache

Setting: Cache.LocalDisk.Enable
Default Value: true
Definition: When true, this setting uses a read cache and upload buffer located on the local machine at /usr/share/nirvanix by default. This setting allows for random writes along with increased performance. A false value will read and write directly with the cloud.
Example: Cache.LocalDisk.Enable=true

Setting: Cache.BasePath
Default Value: /usr/share/nirvanix
Definition: This is the base path containing read cache and upload buffer files. Pending SDN operations are also stored here. Do NOT change this setting without ensuring you have no pending operations or uploads. This can be verified with the Upload Monitor.
Example: Cache.BasePath=/usr/share/nirvanix

Setting: UpdateService.BufferLimit.MB
Default Value: 25%
Definition: The buffer limit can be set in megabytes (example: "100") or as a percentage (example: "25%") available of the designated partition capacity. If this limit is exceeded when writing to the buffer, all files attempting to write will fail.
Example: UpdateService.BufferLimit.MB=25%

Setting: Cache.Persistence.MB
Default Value: [The UpdateService.BufferLimit.MB value]
Definition: The maximum amount of read cache storage in megabytes that is persisted. Anything beyond that will be removed from cache with the least recently accessed files removed first.
Example: Cache.Persistence.MB=1000

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

6. Modules and Functional Descriptions

6.1 Upload Buffer & Read Cache

CloudNAS performs reads and writes on local disk to greatly increase performance. It utilizes both an upload buffer and a read cache. The upload buffer is file data that has been written locally and is pending upload to the SDN. The read cache is SDN file data that is accessible locally for read operations.

All application write requests to CloudNAS are written to the upload buffer and can be read immediately as read cache. After an application writes and closes a file, the file is queued for upload to the SDN as a background task. The upload queue may contain many files. You can use Upload Monitor to see an instant snapshot of this.

The UpdateService.BufferLimit.MB configuration sets the total maximum size of all file data that can be queued for upload. If this limit is reached, any application create file requests will succeed but write requests will return back a failure code.

All pending SDN operations are saved to local hard disk. This is to provide protection in case the machine crashes, powers off, or you stop CloudNAS before it is able to process all of your operations. The next time CloudNAS starts, all pending and partially completed SDN operations will resume.

As you read a file from the SDN, the file data is stored locally in the read cache. This allows follow up reads to be done at local disk speed. The configuration option “Cache.Persistence.MB” sets the size of the cache. When the cache is full, files that have been completely uploaded to the SDN may be removed. Files are removed based on last accessed time with older files getting flushed first. Files in the upload buffer are not affected by this setting.

6.2 Cache Disabled and Write Operations

In this mode, your application may read from any offset within a file when reading. However, file writes must be sequential and an error will occur on any file seeking activity during a write. If the application you are using to move data into the CloudNAS requires non-sequential read and write, you will need to enable the cache.

6.3 Logging

If enabled, all logging is done via the standard Syslog normally located in /var/log/messages. Logging is sent to the log file for system daemons. It should be viewable by any log viewer and follow Syslog standards.

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

6.4 File Ownership

File owners and groups may be set and enforced just as a standard POSIX file system functions. The owners and groups correspond to the users and groups local to the system running Nirvanix CloudNAS.

6.5 File Permissions

Permissions for files and directories also may be set and enforced as expected of a POSIX file system.

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

7. Known Issues

The following are known issues for this release.

- File/Folder modifications made outside of a specific CloudNAS instance will not be immediately reflected in that instance. This is the result of caching optimizations that improve CloudNAS performance. Modifying data directly from the Nirvanix API or from other clients, including other CloudNAS instances, may create inconsistencies and cause problems for CloudNAS as well as applications using it to access the SDN.
- Data that is encrypted using Nirvanix CloudNAS can only be decrypted using Nirvanix CloudNAS.
- Restoring a server which has a CloudNAS installation and/or corresponding cache directory can cause data corruption on the Nirvanix SDN. The CloudNAS cache should be cleared before restarting the service.
- Hard links are not supported.
- Extended attributes are not supported.
- Modification time is implicitly updated by CloudNAS after a file has been written to and closed. Access time is not updated after each file access however both access time and modification time can be set explicitly. Change time or ctime is not supported.
- File Locking is not supported.
- Applications that use “Truncate” or “Set File Length” are not supported when in direct mode.
- CloudNAS and SELinux are not compatible.
- The df command will always report 100TB as a default. This value can be set using the Storage.MaxReported.GB configuration setting.
- The Nirvanix SDN is not case sensitive. This means you may not rename a file to new name where only the case differs (e.g. change “SomeFile” to “somefile”).
- Many file systems support and allow for file paths longer than the SDN supported 512. In cases where this occurs in the CloudNAS, you may receive a “Stale NFS File Handle” error.

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

- Certain file systems and programs do not allow you to use a file/folder with name of length 256 characters. You may experience issues if you exceed their respective limits.

  Here are some examples:

  - Many Windows programs will restrict the whole file path (not just the filename) to the MAX_PATH variable of 256 chars. See the following Microsoft Knowledge Base article for more details: http://support.microsoft.com/kb/177665

  - The max filename length allowed in NFS is 255 chars. Hence if there are files/folders with the name of length 256 chars then the respective folder cannot be listed in the NFS.

  - The max filename length allowed by Linux ext3 is 255 chars, hence if you try to copy a file/folder whose name is of length 256 chars the copy operation will fail.

- CloudNAS designates a directory in the cache for each file that it accesses. The popular Linux file system ext3 has a 32,000 sub-directory limit. CloudNAS will begin to fail if the cache is placed on an ext3 file system and that limit is exceeded. Use of XFS or a similar file system without such limitations will avoid this issue.

- If you receive a “CRITICAL” error that contains this message:

  System.ArgumentOutOfRangeException: Non-negative number required.

  This means that you are overloading your Internet connection with a very large chunk size. This should only occur when you have explicitly changed the UpdateService.PartLength.MB parameter to something higher than the default of 10 MB. To resolve the issue, lower the PartLength or remove the parameter from your configuration file (so the default will be used).

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

8. Troubleshooting

The following information may be helpful in diagnosing and resolving common problems. Some items may touch multiple categories but will only appear under the category that is most appropriate.

8.1 Activation

- CloudNAS will not start
  Check the network connection and make sure you are able to connect the Nirvanix domains outlined under *System Requirements*.

- Cannot ping required Nirvanix domains
  Please check the firewall settings for the machine hosting CloudNAS and for the network connecting it to the internet. You may need to update the firewall settings to allow access to the specified domains and ports defined under *System Requirements*.

8.2 Performance

- Slow uploads to the SDN
  Often this is caused by network connection problems or server configurations.

- High CPU utilization
  CloudNAS was created for installation on a dedicated server and to transfer data to the Nirvanix SDN as fast as possible. Because transferring of data requires use of the CPU, high-bandwidth connections are likely to see higher CPU utilization. Use of the encryption feature will further increase CPU load.

- File system limitations
  The popular Linux file system ext3 has a 32,000 sub-directory limit which can cause problems for the CloudNAS cache mechanism. Use of XFS or a similar file system without such limitations will resolve this issue.

8.3 File Access

- Copying files to CloudNAS fails
  The Nirvanix SDN does not allow the use of certain characters in the file path. To avoid any conflicts when running asynchronously, CloudNAS filters for and rejects any invalid paths. It will reject paths with leading or trailing spaces in the file or folder name, or with any of the following characters: \ / : ? | * " < >

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

8.4 Mounting / Unmounting

- Cannot mount drive or share
  - Verify that the CloudNAS was successfully unmounted, as follows:

    ```
    ps aux |grep cloud |grep -v grep
    ```

    If this command returns any processes, you will need to kill the CloudNAS process manually:

    ```
    kill -9 pid
    ```

    Where the pid is the process ID for the CloudNAS. This is generally in the second column of the output from:

    ```
    ps aux
    ```

  - Verify that the FUSE module is loaded:

    ```
    lsmod |grep fuse
    ```

    If this command does NOT return any output, run the following command:

    ```
    modprobe fuse
    ```

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

Appendix I – NFS Setup

The default kernel-based NFS server does not work with Nirvanix CloudNAS. As an alternative you can use unfs3. The configuration and setup for unfs3 is the same as the kernel-based NFS, only use the unfs3 daemon in place of the kernel-based NFS daemon.

Package	Project Site	Recommended Hosted Package
unfs3	http://unfs3.sourceforge.net/	32-bit http://services.nirvanix.com/Mounts/NirvanixProjects/RH5/unfs3-0.9.20-1.el5.rf.i386.rpm 64-bit http://services.nirvanix.com/Mounts/NirvanixProjects/RH5-64/unfs3-0.9.20-1.el5.rf.x86_64.rpm

After downloading and installing UNFS3, the standard NFS setup applies here, specifically:

Add a line to /etc/exports on the server to export the /nirvanix directory. For example:

/nirvanix (rw,sync,no_root_squash,fsid=2)

This option will export the /nirvanix directory to all users and provide read/write permissions. Note that unlike the standard NFS syntax, you may **not** use an asterisk to represent all hosts; instead, you should leave the host blank. For more information on /etc/exports syntax for UNFS3, you may do a “man unfsd” on the command line after installation.

Start (or restart) the unfsd daemon.

On the client machine, you can mount the directory like so:

mount {cloudnashost}:/mnt/nirvanix

where “cloudnashost” is the hostname or IP address of the machine where CloudNAS and UNFS3 have been installed and /mnt/nirvanix is a directory that you have set up previously as the mount point for CloudNAS.

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

Appendix II – Advanced Configuration

These settings are not normally required. Please contact Nirvanix if you intend to use any of them.

Buffer & Cache

Setting: UpdateService.PartLength.MB
Default Value: 10
Definition: The segment of file dedicated each upload segment. File segments will be uploaded sequentially to the upload buffer.
Example: UpdateService.PartLength.MB=10

Setting: Cache.DirectoryStale.Mins
Default Value: 10080
Minimum Value: 4
Definition: The time in minutes CloudNAS waits to remove a directory from cache and eventually retrieve from the Nirvanix Web Services. The default value is equivalent to 7 days. The lower this value, the more often directory information is refreshed from the Nirvanix SDN.
Example: Cache.DirectoryStale.Mins=10080

Setting: Cache.DirectoryExpiration.Mins
Default Value: none
Minimum Value: 4
Definition: The time in minutes CloudNAS waits to remove a specified directory from cache and eventually retrieve from the Nirvanix Web Services. Directories and cache expiration are displayed in a comma and double slash delimited format.
Example: To specify a cache expiration for folder /child1/foo of 5 minutes and folder /child2/bar of 6 minutes, you would specify the following: Cache.DirectoryExpiration.Mins=/child1/foo//5,/child2/bar//6

Logging

Setting: Log.Syslog.Identity
Default Value: CloudNAS
Definition: The syslog name to identify this file system.
Example: Log.Syslog.Identity=CloudNAS

Setting: Log.Syslog.Facility
Default Value: DAEMON
Definition: The syslog facility where logging occurs when enabled.
Example: Log.Syslog.Facility=DAEMON

Permissions & Ownership

Setting: Permissions.DefaultOverride
Default Value: false
Definition: A true value will use the default permissions for all files.
Example: Permissions.DefaultOverride=false

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.

Setting: Permissions.DefaultDir
Default Value: drwxrwxrwx
Definition: The default permissions to use for directories without permission set. Only applies if Permissions.DefaultOverride is set to true.
Example: Permissions.DefaultDir=drwxrwxrwx

Setting: Permissions.DefaultFile
Default Value: -rw-rw-rw-
Definition: The default permissions to user for files without permissions set.
Example: Permissions.DefaultFile=-rw-rw-rw-

Setting: Permissions.RootNode
Default Value: dr-xr-xr-x
Definition: The permission to use for the root inode. These are the permissions used for "/nirvanix".
Example: Permissions.RootNode=dr-xr-xr-x

Other

Setting: Storage.MaxReported.GB
Default Value: 102400
Definition: The maximum reported size of the CloudNAS volume. This can be viewed using the df command.
Example: Storage.MaxReported.GB=102400

Setting: UploadMonitor.MaxCompletedUploadsHistory
Default Value: 50
Definition: This is the maximum number of completed uploads displayed with the most recently completed displayed first. The minimum value is 1 and the maximum value is 200.
Example: UploadMonitor.MaxCompletedUploadsHistory=50

© 2008 – 2010 Nirvanix, Inc. All Rights Reserved.